Name: Purnima Kabadwal 5.1

Sec (9): Roll- 07 TUTORIAL -5
No

BFS	DFS
· Stands for Breadth first search.	· Stands for Depth fior search.
· DFS uses queue to find the shortest path.	· It uses stack to find shortest path.
· BFS is better when target is closer to source.	· DFS is better when target is far from source.
· As BFS consider all neighbours so it is not suitable for dicisions tree used in puzzle games.	· DFS is more suitable for Decision tree. As with one decision we need to traverse further to argument the decision. If we reach conclusion
· BFS is slower than DFS	

Application of DFS:

- Using DFS we can find path b/w two vertices
- We can perform topological sorting which is used to scheduling jobs.
- We Can use DFS to detect cycles.
- Using DFS, we can find strongly connect components of graph.

Application of BFS :

- BFS may also used to detect cycles.
- finding shortest path and minimal spanning tree in unweighted graph.
- In networking finding a route for packe transmission.
- Finding a route through GPS navigation system.

Question 2.

Breadth first search (BFS) uses Queue data structure. In BFS you mark any node in the graph as source node and start traversing from it. BFS traverses all the nodes in the graph and keeps dropping them as completed. BFS visited an adjacent unvisited node, marks it as done and insert it into Queue.

DFS uses stack data structure because DFS traverse a graph in a depth ward motion and uses a stack to remember to get to next vertex to start a search, when a dead end occurs. in any iteration.

Ques - 3 :-

Sol^{n} sparse graph :- A graph in which the number of edges is much less than the possible number of edges.

5.2.

Dense Graphs : A dense Graph is a graph in which the number of edges is close to the maximal no. of edges.

→ If the agraph is ~~bottes~~ sparse, we should store it as list of edges.

Alternatively if a graph is dense, we should store it as adjacency matrix.

Ques4

Sol · DFS can be used to detect cycle in a graph.

· DFS for a connected graph produces a tree. there is a cycle in a graph only if there is an edge present in the graph. A back edge is an edge that is from a node to itself or one of its ancestor in the tree produced by DFS.

BFS can be also used to detect cycles. Just perform BFS while keeping a list of previous nodes at each node visited or else constructing a tree from starting node. If I visit a node that is already marked by BFS, I found a cycle

Ques 5

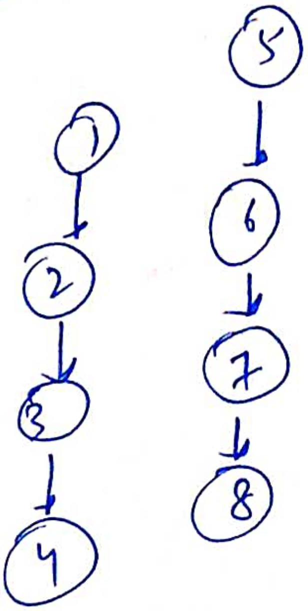
Ans 5 Dijoint set Data structure

- It allows to find out whether the two elements are in the same set or not efficiently.

- A disjoint set can be defined as the subsets when there is no common element between the two sets.

e.g → $s_1 = \{1, 2, 3, 4\}$

$S_2 = \{5, 6, 7, 8\}$



operations performed.

(i) find :

```
int find (int v)
    {  if (v = = parent[v])
           return v;
       return parent[v] = find (parent [v]);
    }
```

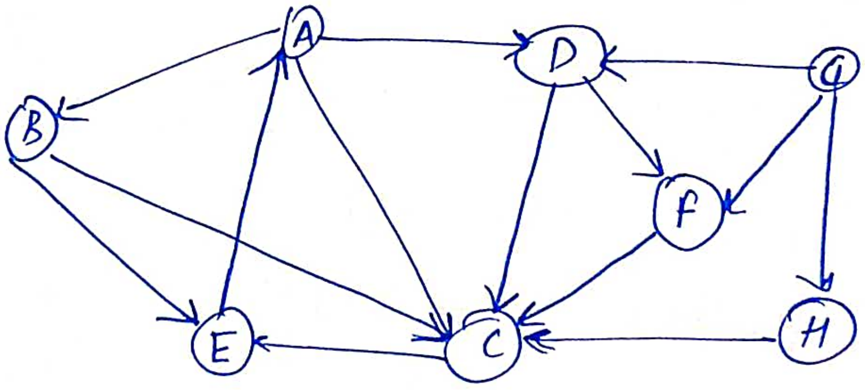
Union :-

```
Void union(int a, int b)
{ a = find (a)
  b = find (b)
  if (a != b)

    { if (size [a] < size [b])
        { swap (a, b) }
      parent [b] = a;
        size [a] += size [b];
    }
}
```

Ques 6 :

Soln



BFS :

Node :	B	E	C	A	D	F
Parent :	A	B	B	E	A	D

Path. B → E → A → D → F

DFS : Node processed B B C E A D F

Stack B CE FE AE DE FE E

Path → B → C → E → A → D → F

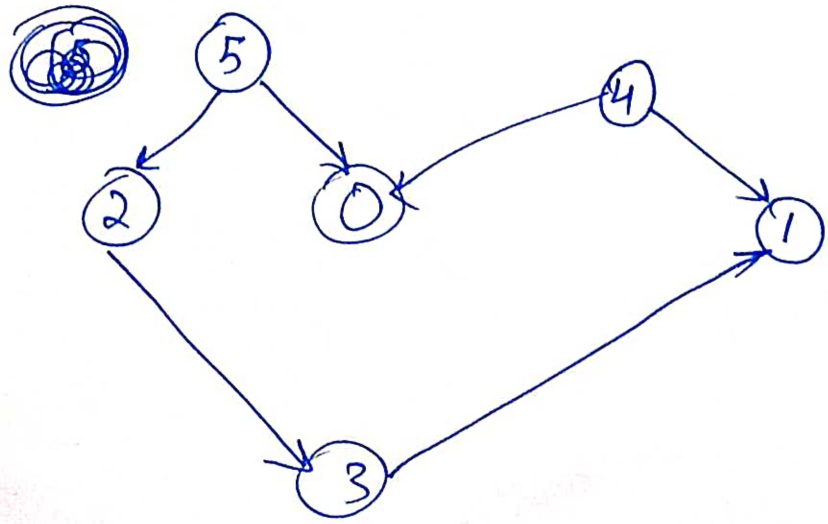
Ques-7

Ans V = {a} {b} {c} {d} {e} {f} {g} {h} {i} {j}

E = {a,b} {a,c} {b,c} {b,d} {e,f}, {e,g}, {h,i}, {j}

(a, b)	{a,b} {c} {d} {e} {f} {g} {h} {i} {j}
(a,c)	{a,b,c} {d} {e} {f} {g} {h} {i} {j}
(b, c)	{a,b,c} {d} {e} {f} {g} {h} {i} {j}
(b,d)	{a,b,c,d} {e} {f} {g} {h} {i} {j}
(e,f)	{a,b,c,d} {e,f} {g} {h} {i} {j}
(e,g)	{a,b,c,d} {e,f,g} {h} {i} {j}
(h,i)	{a,b,c,d} {e,f,g} {h,i} {j}
	{j}

Q.8



5.1

5.4

Adjacency list :-

0 →

1 →

2 → 3

3 → 1

4 → 0, 1

5 → 2, 0

0	1	2	3	4	5
F	F	F	F	F	F

f → false.

Stack → empty.

Step 1 - Topological sort [0], visited [0] = true.

Stack

step 2: Topological sort (1), visited [1] = true

Stack

0	1

step 3: Topological sort (2), visited [2] = true

Topological sort (3), visited [3] = true.

Stack

0	1	3	2

step 4:
stack

0	1	3	2	4

step 5:- Stack

0	1	3	2	4	5

step 6: Print all elements of stack from top to bottom. → 5 4 2 3 1 0.

Question 9.

Ans. Algorithm that uses Priority Queue:-

(i) Dijkstra shortest path algorithm using priority Queue when graph is sorted in the form of list or matrix, priotity queue can be used to extract minimum ~~key node at every step~~ efficieny when implementing Dijktra's Algo.

(ii) Prim's Algorithms:-

It is used to implement prims algorithm to store key of nodes to extract minimum key node at every step.

(iii) Data compression: It is used in Huffman's algoritm code which is used to compress data.

Q. 10 Ans

Min Heap	Max Heap.
In min. heap the key present at root must be less than or equal to among the keys present at all its children	In max heap the key present at the root must be greater or equal to the key present at all its childrens.
· uses the ascending priority.	· uses descending priority
· The minimum key present at the root node.	· The maximum key present at the root node.